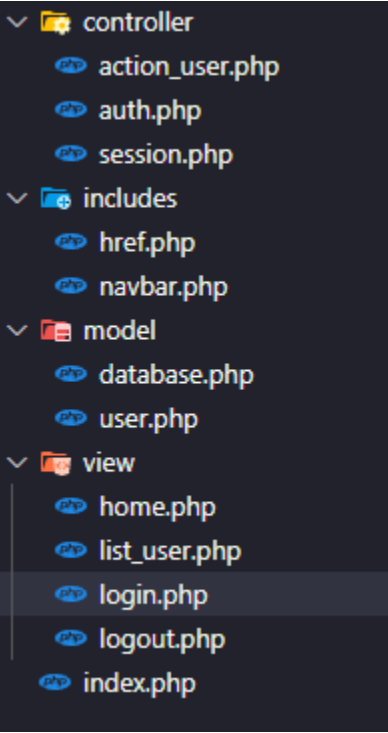


ACTIVITY LOGIN / SESSION / AUTH / HOME



Creating a login system using PHP with MySQLi in an Object-Oriented Programming (OOP) style involves several steps. Below is a structured guide to help you set up a basic login system with session management and authentication.

In this image Your CRUD folder looks like. Add this file on their designation

Controller: auth.php, session.php

Includes: navbar.php

View: home.php , login.php , logout.php

STEP 1: Create a login.php / Design and Form

view/login.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>LOGIN</title>
  <?php include 'includes/href.php'; ?>
</head>
<body>
  <div class="container-fluid">
    <div class="row text-center my-5">
      <h1 class="text-center">LOGIN</h1>
    </div>
    <div class="row">
      <div class="col-md-4 offset-md-4 my-5">
        <div class="card border-primary">
          <div class="card-body p-5">
            <?php
              if(isset($_GET['error'])){ ?>
                <div class="alert alert-danger alert-dismissible fade show" role="alert">
                  <?php echo $_GET['error']; ?>
                  <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
                </div>
              <?php } ?>
              <form action="controller/auth.php" method="POST">
                <div class="form-group my-2">
                  <label for="username">Username</label>
                  <input type="text" name="username" class="form-control" required>
                </div>
                <div class="form-group my-2">
                  <label for="password">Password</label>
                  <input type="password" name="password" class="form-control" required>
                </div>
                <div class="form-group my-2">
                  <button type="submit" name="login" class="btn btn-primary">Submit</button>
                </div>
              </form>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

STEP 2: Create your home.php when the login success it will go first on home page
(You can alter or design of it and use Bootstrap).

view/home.php

```
<!DOCTYPE html>
<html Lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HOME</title>
    <?php include '../includes/href.php'; ?>
</head>
<body>
    <?php include '../includes/navbar.php'; ?>
    <div class="container-fluid">
        <div class="row text-center my-5">
            <h1>HOME</h1>
        </div>
        <div class="row text-center my-5">
            <h1>WELCOME TO MY PAGE</h1>
            <h2><?php echo $user_login['fname'].' '.$user_login['lname']; ?></h2>
        </div>
    </div>
</body>
</html>
```

STEP 3: Create your navigation bar on navbar.php
includes/navbar.php

```
<nav class="navbar navbar-expand-lg bg-primary data-bs-theme="dark">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">LOGO</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
            data-bs-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup"
            aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
            <div class="navbar-nav">
                <a class="nav-link active" aria-current="page" href="home.php">Home</a>
                <a class="nav-link" href="list_user.php">User Management</a>
            </div>
        </div>
        <div class="d-flex">
            <span class="navbar-text text-white me-3"><?php echo $user_login['username']; ?></span>
            <a class="btn btn-danger" href="logout.php">logout</a>
        </div>
    </div>
</nav>
```

STEP 4: These two functions should be added to your model/user.php. The logic for your login is stored in this function, which verifies that the username and password you entered in login.php are correct. If there is a data match on your login, it will process in your database.

model/user.php

```
1 reference | 0 overrides
public function login($username, $password): bool {
    $sql = "SELECT * FROM tbl_user WHERE username = ? AND password = ?";
    $stmt = $this->conn->prepare(query: $sql);
    $stmt->bind_param(types: "ss", var: &$username, vars: &$password);
    $stmt->execute();
    $result = $stmt->get_result();
    return $result->num_rows > 0;
}

1 reference | 0 overrides
public function getUserId($username, $password): mixed {
    $sql = "SELECT user_id FROM tbl_user WHERE username = ? AND password = ?";
    $stmt = $this->conn->prepare(query: $sql);
    $stmt->bind_param(types: "ss", var: &$username, vars: &$password);
    $stmt->execute();
    $result = $stmt->get_result();
    return $result->fetch_assoc()['user_id'];
}
```

STEP 5: Make sure your controller folder contains an auth.php file. Start the class with variable \$user, add an if condition for \$_POST['login'], then use session_start() to record the login ID and include the model/user.php directory. Utilize the login method; if it is successful, the user ID is obtained and stored on \$_SESSION['user_id']. This session is then used on all sites to identify the user that signs in.

controller/auth.php

```
<?php
session_start();
include '../model/user.php';
$user = new User();

if($_SERVER['REQUEST_METHOD'] == 'POST'){
    if(isset($_POST['login'])){
        $username = $_POST['username'];
        $password = $_POST['password'];

        if($user->login(username: $username, password: $password)){
            $_SESSION['user_id'] = $user->getUserId(username: $username, password: $password);
            header(header: 'location: ../view/home.php');
            exit();
        }else{
            header(header: 'location: ../index.php?error=Invalid username or password');
            exit();
        }
    }
}
...
?>
```

STEP 6: Make a file on controller/ called session.php. If the \$_SESSION['user_id'] is not empty or signed in, this file verifies it. It will return to location index.php if it is empty. If not empty it getUserById on Class User

controller/session.php

```
<?php
session_start();
if(!isset($_SESSION['user_id'])){
    header(header: 'location: ../index.php');
    exit();
}

include '../model/user.php';
$user= new User();
$user_login = $user->getUserById(id: $_SESSION['user_id']);

?>
```

STEP 7: To prevent accessing this website without logging in, add the controller/session.php in view/home.php.

```
<?php
include '../controller/session.php';
?>
```

STEP 8: Create a file named logout.php in view. To stop someone from visiting the website without signing in, it will end the session and return to index/login.

view/logout.php

```
<?php
session_start();
session_destroy();
header(header: 'location: ../index.php');

?>
```

STEP 9: on your index.php include the view/login.php

```
<?php
include 'view/login.php';

?>
```

STEP 10: That’s it you have a login system, Try and check if working properly

Note: There will be a problem with list_user.php. Since you have previously created a Class User on session.php and have a variable called \$user, remove this code from the top of your page. It creates copies.

```
include '../model/user.php';
$user = new User();
```